Web Images Videos Maps News Shopping Gmail more ▾ Scholar Preferences | Sign in

Google scholar genetic algorithm optimize compiler benchmar Search Advanced Scholar Search

Scholar Articles and patents - 2004 include citations ☐ Create email alert Results **1** - **10** of about **254**. (**0.12** sec)

Combined selection of tile sizes and unroll factors using iterative compilation
T Kisuki, PMW Knijnenburg... - Parallel Architectures ... 2002 - ieeexplore.ieee.org
... transformations evaluated so far and decides which trans- formations have to be applied next using a search **algorithm ...** MT1 **Compiler** TDL-Flies ... We have imple- mented several search algorithms, including a **genetic** algo- rithm, simulated annealing, pyramid search, window ...
Cited by 140 - Related articles - All 6 versions

Fast searches for effective optimization phase sequences
P Kulkarni, S Hines, J Hiser, D Whalley... - Proceedings of the ..., 2004 - portal.acm.org
... the percentage improvement that we obtained for the SPARC when **optimizing** for speed ... The baseline measures were obtained using the batch VPO **compiler**, which iteratively ... include additional optimization phases that were not previously exploited by the **genetic algorithm**. ...
Cited by 78 - Related articles - BL Direct - All 4 versions

Compiler based exploration of DSP energy savings by SIMD operations [PDF] from york.ac.uk
M Lorenz, P Marwedel, T Dräger... - Proceedings of the ..., 2004 - portal.acm.org
... for an overview see eg [1]). However, to obtain an energy-efficient system, **optimizing** the software ... In [12] we have published **compiler** optimization techniques with the aim of minimizing the ... into the GeLIR-code and is then compacted by reusing the **genetic algorithm** driven code ...
Cited by 21 - Related articles - All 24 versions

Iterative compilation
P Knijnenburg, T Kisuki ... - Embedded processor design ..., 2002 - Springer
... We have implemented several search algorithms, including a **genetic algorithm**, simulated annealing, pyramid search, window search and random search. ... Driver List of Transformations MT1 **Compiler** TDL−Files F77 ... 3 **Benchmarks** and Platforms ...
Cited by 21 - Related articles - BL Direct - All 6 versions

Optimization parameter selection by means of limited execution and **genetic** algorithms
Y Che, Z Wang... - Advanced Parallel Processing Technologies, 2003 - Springer
... Niclos G. Fournier.: Enhancement of an Evolutionary **Optimizing Compiler**, Ph.D Thesis ... Shuvra S. Bhattacharyya.: A Joint Power/Performance Optimization **Algorithm** for Multiprocessor ... et al.: Automatic parallel I/O performance optimization using **Genetic** Algorithms, Proceedings ...
Cited by 6 - Related articles - BL Direct - All 4 versions

Combined selection of tile sizes and unroll factors using iterative compilation
PMW Knijnenburg, T Kisuki... - The Journal of ..., 2003 - Springer
... to optimization has been to have a human expert hand-**optimize** the application, a ... **Genetic algorithm**: **Genetic** Algorithms are modeled on natural evolution processes and manipulate individuals in a ... to the **target** architecture, we used the native Fortran77 or g77 **compiler** with full ...
Cited by 10 - Related articles - BL Direct - All 5 versions

Adaptive java optimisation using instance-based learning
S Long ... - Proceedings of the 18th annual international ..., 2004 - portal.acm.org
... Suppose, for program D in Fig- ure 1, the **algorithm** locates from its most similar ... approach would be to initially allow an existing high level restructurer to **optimise** each new ... technique described in this paper was imple- mented in a Java restructuring **compiler** and evaluated on ...
Cited by 25 - Related articles - All 10 versions

Energy aware compilation for DSPs with SIMD instructions [PS] from tu-dortmund.de
M Lorenz, L Wehmeyer... - ... systems: software and **compilers** ..., 2002 - portal.acm.org
... and thus has an essential impact on the optimization progress of the **genetic algorithm**. ... by 7%, whereas the number of memory accesses did not change for these **benchmarks**. ... The growing use of DSPs in embedded systems necessitates **optimizing compilers** supporting special ...
Cited by 27 - Related articles - BL Direct - All 20 versions

Phase coupled code generation for DSPs using a **genetic algorithm** [PDF] from date-conference.com
M Lorenz... - Proceedings of the conference on Design ..., 2004 - portal.acm.org
... Table 1: **Benchmark** characteristics #CSE **benchmark** #CSEs uses CPU[s] cm complex multiply 4 8 19 ... The growing use of DSPs in embedded systems necessitates **optimizing compilers** which are ... In this paper we have presented a **genetic algorithm** driven code generator which ...
Cited by 7 - Related articles - All 14 versions

Statistical selection of **compiler** options
RPJ Pinkers, PMW Knijnenburg... - ..., and Simulation of ..., 2004 - ieeexplore.ieee.org
... In Section 4, we propose our interactive **algorithm** for enabling options and in Section 5 we discuss our experi- mental framework. ... In our case, columns correspond to **compiler** options and each row is a particular **compiler** setting that can be used to **optimize** a program. ...
Cited by 21 - Related articles - All 6 versions

☐ Create email alert

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

genetic algorithm optimize compiler Search

Go to Google Home - About Google - About Google Scholar

Web Images Videos Maps News Shopping Gmail more ▾ Scholar Preferences | Sign in

Google scholar (tune OR tailor) compiler target platform bench Search Advanced Scholar Search

Scholar Articles and patents anytime include citations Create email alert Results **1** - **10** of about **4,430**. (**0.06** sec)

[PDF] **Compiler** support of the workqueuing execution model for Intel SMP architectures [PDF] from caspur.it
E Su, X Tian, M Girkar, G Haab, S Shah ... - European Workshop on ..., 2002 - caspur.it
... We also present preliminary performance results of a set of **benchmarks** and applications measured on ... propagation, partial redundancy elimination (PRE) and partial dead store elimination (PDSE) • **Target**-specific optimizations ... **Compiler** Support of the Workqueuing Execution ...
Cited by 42 - Related articles - View as HTML - All 4 versions

Effectiveness of cross-**platform** optimizations for a Java just-in-time **compiler** [PDF] from ibm.com
K Ishizaki, M Takeuchi, K Kawachiya... - Proceedings of the ..., 2003 - portal.acm.org
... At the same time, it is desirable to **tune** the performance for the **target** architecture to ... Finally, we perform code emission to generate the machine instructions for the **target** architecture in cooperation ... Version 1.4.0. The threshold in the interpreter to initiate the JIT **compiler** was set ...
Cited by 27 - Related articles - BL Direct - All 12 versions

[CITATION] A library-based **compiler** to execute MATLAB programs on a heterogeneous **platform**
A Nayak, M Haldar, A Kanhere, P Joisha... - Proceedings of the ..., 2000 - Citeseer
Cited by 8 - Related articles - All 2 versions

Flexware: a retargetable embedded-software development environment
PG Paulin... - IEEE Design and Test of Computers, 2002 - computer.org
... This flexible design has become the basis for a more recent MPEG4 codec **platform** that mobile applications use. ... Finally, C **compiler** developers may use the tool to fine-**tune compiler** optimizations for the **target** architecture. ...
Cited by 17 - Related articles - BL Direct - All 8 versions

NPCryptBench: a cryptographic **benchmark** suite for network processors [PDF] from cornell.edu
Y Yue, C Lin ... - ACM SIGARCH Computer Architecture News, 2006 - portal.acm.org
... Finally, we propose several optimizations to **tune** the **benchmark**. ... Section 4 and section 5 present **compile**-time and run- time characteristics of NPCryptBench on Intel ... algorithms and describe rules that we follow when implementing the **benchmark** on a proposed **target platform**. ...
Cited by 5 - Related articles - BL Direct - All 12 versions

Code size reduction by **compiler** tuning
M Haneda, P Knijnenburg... - Embedded Computer Systems: ..., 2006 - Springer
... In previous work, we have used so-called the main effect of **compiler** options to **tune** the **compiler** ... unswitch-loops 49 old-unroll-loops 50 branch-**target**-load-optimize 51 branch-**target**-load-optimize2 ... No complex new transformations or other adaptation of the **compiler** are needed ...
Cited by 4 - Related articles - BL Direct - All 3 versions

An evaluation of global address space languages: Co-Array Fortran and Unified Parallel C [PDF] from rice.edu
C Coarfa, Y Dotsenko, J Mellor-Crummey... - Proceedings of the ..., 2005 - portal.acm.org
... library called GASNet [2]. The GASNet library is optimized for a variety of **target** archi- tectures ... options: -override limits -O3 -g -tpp2 2back-end **compiler** options: -fast -O5 -**tune** host -intrinsics 38 ... IRIX64 V6.5, the MIPSpro **Compilers** V7.4 and the Berkeley UPC **compiler** V2.0.14 ...
Cited by 50 - Related articles - All 20 versions

Co-array Fortran performance and potential: An NPB experimental study [PDF] from tamu.edu
C Coarfa, Y Dotsenko, J Eckhardt... - ... and **Compilers** for ..., 2004 - Springer
... Although the language provides shared-memory semantics, the **target** architecture may not. ... On a hardware shared memory **platform**, the transformation is relatively straightfor- ward since ... 3 was used along with the override-limits option to prevent the **compiler** from automatically ...
Cited by 37 - Related articles - BL Direct - All 21 versions

Automatic **benchmark** generation for cache optimization of matrix operations [PS] from clemson.edu
J McCalpin... - Proceedings of the 33rd annual on ..., 1995 - portal.acm.org
... character- istics of the hardware and software, as well as to **tune compiler** heuristics and ... may simply want to obtain the best possi- ble performance on each **target platform** with the ... that were designed to provide automatic generation and execution of **benchmark** programs from ...
Cited by 3 - Related articles - All 3 versions

Rapidly selecting good **compiler** optimizations using performance counters [PDF] from cmu.edu
J Cavazos, G Fursin, F Agakov... - ..., 2007. CGO'07. ..., 2007 - ieeexplore.ieee.org
... Using the model Given a new **target benchmark**, we first extract the perfor- mance counter features x by running the **benchmark**. This requires 3 runs of the **benchmark**. ... These **benchmarks** are used by PathScale to **tune** their **compiler** suite. ...
Cited by 81 - Related articles - All 23 versions

Create email alert

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

(tune OR tailor) compiler target platf Search

Go to Google Home - About Google - About Google Scholar

Web Images Videos Maps News Shopping Gmail more ▾ Scholar Preferences | Sign in

Google scholar target (platform OR architecture) specific comp Search Advanced Scholar Search

Scholar Articles and patents anytime include citations Create email alert Results **1** - **10** of about **34,100**. (**0.03** sec)

Parallel programming using skeleton functions [PDF] from ic.ac.uk
J Darlington, A Field, P Harrison, P Kelly... - PARLE'93 Parallel ..., 1993 - Springer
... These primitives provide a **platform** on which skeletons describing SIMD computations can be defined. ... of the specification to take advantage of the particular characteristics of an **architecture** without compromising ... J15] as the source language and using C as the **target** language ...
Cited by 388 - Related articles - BL Direct - All 15 versions

Meta optimization: improving **compiler heuristics** with machine learning [PDF] from colostate.edu
M Stephenson, S Amarasinghe, M Martin... - ACM SIGPLAN ..., 2003 - portal.acm.org
... The algorithm stops merging paths when it has consumed the **target** archi- tecture's estimated resources. ... Trimaran is an integrated **compiler** and simulator for a pa- rameterized EPIC **architecture**. Table 3 details the **specific architecture** over which we evolved. ...
Cited by 144 - Related articles - BL Direct - All 26 versions

Compiler optimization-space exploration [PDF] from ulp.org
S Triantafyllis, M Vachharajani... - ... 2003. CGO 2003. ..., 2003 - ieeexplore.ieee.org
... iterative compilation works are limited to **specific** ar- chitectures, limited to **specific** optimizations, or ... The Itanium proces- sor makes a good **target architecture** since explicitly parallel machines depend ... Electron is among the best **compilers** for the Itanium **platform**, thus providing a ...
Cited by 154 - Related articles - All 22 versions

Debugging system with portable debug environment-independent client and non-portable **platform-specific** server
LL You, N Rajgopal ... - US Patent 5,815,653, 1998 - Google Patents
... DEBUGGING SYSTEM WITH PORTABLE DEBUG ENVIRONMENT-INDEPENDENT CLIENT AND NON-PORTABLE **PLATFORM- SPECIFIC** SERVER 5 ... trans- lation process varies based on the **compiler** program itself, the processor **architecture**, the **target** runtime execution ...
Cited by 71 - Related articles - All 2 versions

Address calculation for retargetable compilation and exploration of instruction-set architectures [PDF] from york.ac.uk
C Liem, P Paulin... - ... of the 33rd annual Design Automation ..., 1996 - portal.acm.org
... the **target** can be fed ... Parallelization (compaction) is left for the back-end **architecture compiler**. ... In our experience, these items are common in an embedded sys- tem development methodology, where firmware is simulated on a desk-top **platform** before being used in the field. ...
Cited by 78 - Related articles - BL Direct - All 10 versions

The Chinook hardware/software co-synthesis system [PDF] from kfupm.edu.sa
PH Chou, RB Ortega... - Proceedings of the 8th ..., 1995 - portal.acm.org
... Chinook does not compile code to the **target** processor(s). It assumes not only the ... heterogeneous as cost and modularity concerns drive designers to tailor processors to **specific** functions ... We modeled this **architecture** with three handlers, one for the processor re- quests, one for ...
Cited by 198 - Related articles - Library Search - All 16 versions

Statistical selection of **compiler options**
RPJ Pinkers, PMW Knijnenburg... - ..., and Simulation of ..., 2004 - ieeexplore.ieee.org
... is (almost) fully automatic and requires (almost) no knowledge about the **compiler** or the **target architecture**. ... benchmarks when compiled with GCC 2.6.3 and ran on the SimpleScalar **platform**. ... This shows that tuning **compiler** settings for a **specific** application can be worthwhile. ...
Cited by 21 - Related articles - All 6 versions

Genetic programming applied to **compiler heuristic** optimization [PDF] from unb.br
M Stephenson, UM O'Reilly, M Martin... - Genetic ..., 2003 - Springer
... Page 6. Genetic Programming Applied to **Compiler Heuristic** Optimization 243 ... Trimaran's **compiler**, which is called IMPACT, performs code profiling. Table 3 details the **specific architecture** over which we evolved. This model is similar to Intel's Itanium **architecture**. ...
Cited by 22 - Related articles - BL Direct - All 12 versions

A machine learning approach to automatic production of **compiler** heuristics [PDF] from ucsb.edu
A Monsifrot, F Bodin... - Artificial Intelligence: Methodology, ..., 2002 - Springer
... revision, but also at new implementations of the **target** Instruction Set **Architecture**, a new ... a learning process which adapts to new **target** architectures or new **compiler** features ... an abstract loop representation we showed that decision trees that provide **target specific** heuristics for ...
Cited by 32 - Related articles - BL Direct - All 11 versions

Automatic selection of **compiler options** using non-parametric inferential statistics
M Haneda, PMW Knijnenburg... - ... 2005. PACT 2005. ..., 2005 - ieeexplore.ieee.org
... that the best opti- mization sequence depends on both the application as well as the **target architecture**. ... to set back-end com- piler switches for any application and **architecture** automat- ically. ... As is well known, each application requires its own **specific** setting of these **options** to ...
Cited by 32 - Related articles - All 6 versions

Create email alert

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

target (platform OR architecture) spe Search

Go to Google Home - About Google - About Google Scholar

Web Images Videos Maps News Shopping Gmail more ▾ Scholar Preferences | Sign in

Google scholar training set (tailor OR tune) compiler Search Advanced Scholar Search

Scholar Articles and patents ▾ anytime ▾ include citations ▾ Create email alert Results **1** - **10** of about **13,100**. (**0.17** sec)

Adaptive optimizing **compilers** for the 21st century [PDF] from ncsu.edu
KD Cooper, D Subramanian... - The Journal of Supercomputing, 2001 - Springer
... particularly important codes, the user may want a version that limits its **training set** to that ... versus compi- lation sequences; restricting the **set** of optimizations to a smaller **set** that has ... computers—often have myriad flags that let a benchmarking specialist hand-**tune** the **compiler's** ...
Cited by 148 - Related articles - BL Direct - All 15 versions

Predicting unroll factors using supervised classification [PDF] from mit.edu
M Stephenson... - Proceedings of the international ..., 2005 - portal.acm.org
... The task of a classifier is to learn how best to map loop characteristics (xi) to the observed labels (yi) using all the examples in the **training set**. While supervised learning is trained offline, the learned classifier can easily be incorporated into a **compiler**. 4.2. ...
Cited by 70 - Related articles - All 19 versions

Rapidly selecting good **compiler** optimizations using performance counters [PDF] from cmu.edu
J Cavazos, G Fursin, F Agakov... - ..., 2007. CGO'07. ..., 2007 - ieeexplore.ieee.org
... values for which enabling the transforma- tion t i leads to improved performance in the **training set** and also ... Note that gathering **training** data and construction of the model is an offline process, that is, it would ... These benchmarks are used by PathScale to **tune** their **compiler** suite. ...
Cited by 61 - Related articles - All 23 versions

Meta optimization: improving **compiler** heuristics with machine learning [PDF] from colostate.edu
M Stephenson, S Amarasinghe, M Martin... - ACM SIGPLAN ..., 2003 - portal.acm.org
... more, by evolving a **compiler's** heuristic over several bench- marks, we can create effective, general-purpose heuristics. The best general-purpose heuristic our system found for hy- perblock formation improved performance by an average of 25% on our **training set**, and 9% on ...
Cited by 144 - Related articles - BL Direct - All 26 versions

MILEPOST GCC: machine learning based research **compiler** [PDF] from inria.fr
G Fursin, C Miranda, O Temam, M Namolaru ... - 2008 - hal.inria.fr
... Drivers for iterative compilation and model **training** ... In an additional **set** of enhancements, a coherent event and data passing mechanism enables external plugins to discover the state of the **compiler** and ... ML drivers to optimize programs and **tune compiler** optimization heuristic ...
Cited by 24 - Related articles - All 23 versions

Genetic programming applied to **compiler** heuristic optimization [PDF] from unb.br
M Stephenson, UM O'Reilly, M Martin... - Genetic ..., 2003 - Springer
... our system found improves the predication al- gorithm by an average of 25% on our **training set**, and 9% on a com- pletely unrelated test **set**. ... **Compiler** writers tediously fine-**tune** priority functions to achieve suitable performance [2]. Priority functions are widely used and tied to ...
Cited by 22 - Related articles - BL Direct - All 12 versions

[CITATION] Sequential minimal optimization: A fast algorithm for **training** support vector machines [PDF] from microsoft.com
J Platt - 1998 - Citeseer
Cited by 353 - Related articles - All 38 versions

Automatic performance model construction for the fast software exploration of new hardware designs [PDF] from pascal-network.org
J Cavazos, C Dubach, F Agakov... - ... on **Compilers**, ..., 2006 - portal.acm.org
... impact of **compiler** optimizations on any new program. As a result, we can drastically reduce the overall simulation time necessary to evaluate tentative architectures and **tune** programs to ... At first, it may be surprising that such a small **training set** size is sufficient to capture such a ...
Cited by 33 - Related articles - All 33 versions

Using machine learning to focus iterative optimization [PDF] from pascal-network.org
F Agakov, E Bonilla, J Cavazos... - ..., 2006. CGO 2006. ..., 2006 - ieeexplore.ieee.org
... This approach is independent of search algorithm, search space or **compiler** infrastructure and scales gracefully with the **compiler** optimization space size. Off-line, a **training set** of programs is iteratively evaluated and the shape of the spaces and program features are modelled. ...
Cited by 136 - Related articles - All 31 versions

Feature selection and policy optimization for distributed instruction placement using reinforcement learning [PDF] from utexas.edu
KE Coons, B Robatmili, ME Taylor... - Proceedings of the ..., 2008 - portal.acm.org
... across a va- riety of applications leave users with little ability to **tune** performance-critical ... target for machine learning because the solution space is large and the **compiler** must make its ... actually very good general solu- tions; the heuristics learned on a **training set** of benchmarks ...
Cited by 9 - Related articles - All 13 versions

Create email alert

Goooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

training set (tailor OR tune) compiler Search

Go to Google Home - About Google - About Google Scholar

Web Images Videos Maps News Shopping Gmail more ▾ Scholar Preferences | Sign in

Google scholar training set loop code fragment complexity Search Advanced Scholar Search

Scholar Articles and patents ▾ anytime ▾ include citations ▾ ✉ Create email alertResults **1** - **10** of about **10,800**. (**0.18** sec)

A compiling genetic programming system that directly manipulates the machine **code**
P Nordin - Advances in genetic programming, 1994 - books.google.com
... These limitations reduce the **complexity** and thus execution time of the individual programs. **...** the **training set**, but that presumably had something in common with the examples in the **training set**. The machine **code** functions, the individuals in the population take a 32 bit integer as **...**
Cited by 211 - Related articles

Visual learning by evolutionary and coevolutionary feature synthesis [PDF] from psu.edu
K Krawiec... - Evolutionary Computation, IEEE ..., 2007 - ieeexplore.ieee.org
... In this way, provides feedback to the search process and closes the learning **loop**. **...** The resulting vectors of features , computed for all images from the **training set** , are the basis for estimating the utility of for recognizing the objects from the **training** data . **...**
Cited by 73 - Related articles - BL Direct - All 19 versions

Hybrid engine for polymorphic shellcode detection [PDF] from shell-storm.org
U Payer, P Teufl ... - Intrusion and Malware Detection and ..., 2005 - Springer
... For the **training** process the Levenberg-Marquardt [10] back-propagation method was used. **...** aad, aam, aas, daa, das 12 jmp 27 clc, cld, cli, clts, clflush 13 inc, dec 28 cbw, cwd, cdq, cdwe 14 **loop**, loope, loopne **...** Further instructions from the X86 **set** were then added to the groups. **...**
Cited by 27 - Related articles - All 15 versions

Data mining static **code** attributes to learn defect predictors [PDF] from psu.edu
T Menzies, J Greenwald... - IEEE Transactions on Software ..., 2007 - computer.org
... More formally,. $$P(H|E)={{P(H)}\over{P(E)}}\prod_iP(E_i|H); $$ ie, given **fragments** of evidence E_i and a **...** or "defect-free") is calculated, given the attributes extracted from a module such as the lines of **code**, the McCabe **...** A learner is then applied to a **training set** built from nin.
Cited by 203 - Related articles - BL Direct - All 18 versions

Tracking down software bugs using automatic anomaly detection [PDF] from psu.edu
S Hangal... - 2002 - computer.org
... 3: Sample **code** from multiprocessor simulator so because DIDUCE did not support disabling these checks as easily at that time, and the overhead was not a significant limitation. We **set** up DIDUCE to use the initial part of each simulation run for **training**, and ignored the **...**
Cited by 372 - Related articles - BL Direct - All 16 versions

[PDF] A map reduce framework for programming graphics processors [PDF] from psu.edu
B Catanzaro, N Sundaram... - Workshop on Software Tools for ..., 2008 - Citeseer
... solver, which has many tight loops with relatively small Map Reduce computations in each **loop**. **...** Working **set** selection using second order information for **training** support vector machines. J. Mach. **...** Fast **training** of support vector machines using sequential minimal optimization. **...**
Cited by 13 - Related articles - View as HTML - All 17 versions

Protein **fragment** clustering and canonical local shapes
CG Hunter... - Proteins: Structure, Function, ..., 2003 - interscience.wiley.com
... A variation on this process is to **loop** through the **fragments** several times, each time increasing the **...** PDB **code** Protein Fold class Resolution (A˚) Size (n) Mean cRMSD (Å) Max cRMSD (Å) **...** Therefore, the **training set** size is a limiting factor when building high- resolution basis sets **...**
Cited by 45 - Related articles - BL Direct - All 4 versions

Use of vector processing to search the Cambridge Structural Database
AHM Thiers, JH Noordik ... - Journal of chemical ..., 1990 - ACS Publications
... **training set**), 216-00-2; 27 (polycyclic **training set**), 195-00-6; 28 (polycyclic **training set**), 222-78 ... op- timized program a significant part of the search time is spent outside our vectorized SCREENS **loop**. **...** The user interfacing to QUEST to **set** up the queries (on the VAX front-end) is **...**
Cited by 2 - Related articles - All 5 versions

Genetic graph programming for object detection
K Krawiec, P Lijewski - Artificial Intelligence and Soft Computing–ICAISC ..., 2006 - Springer
... Only a few contributions [1,2,3,12,14,11,8,9] attempt to close the feedback **loop** of the learning process at the highest (eg, recognition) level, and test the proposed approach in a real-world setting. ... **Training set** Testing **set** Average fitness over all runs 0.9770±0.0133 **...**
Cited by 1 - Related articles - BL Direct - All 2 versions

A meta-heuristic approach to parallel **code** generation [PDF] from psu.edu

B McCollum, PH Corr... - Proceedings of the 5th ..., 2002 - portal.acm.org

... perceptron model to recommend a particular partitioning, selected from a restricted **set**, to apply ... **Training** the neural network requires a representative selection of loops, each of which must ... characteristics to the data partitioning which gives maximum speed up in **loop** execution. ...

Cited by 1 - Related articles - BL Direct - All 8 versions

✉ Create email alert

Gooooooooogle ▶

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

training set loop code fragment compl | Search

Go to Google Home - About Google - About Google Scholar

Web Images Videos Maps News Shopping Gmail more ▾ Scholar Preferences | Sign in

Google scholar loop "training set" OR "test suite" compiler Search Advanced Scholar Search

Scholar Articles and patents ▾ anytime ▾ include citations ▾ □ Create email alert Results **1** - **10** of about **5,900**. (**0.05** sec)

Vectorizing **compilers**: A **test suite** and results
D Callahan, J Dongarra - Proceedings of the 1988 ACM/ ..., 1988 - portal.acm.org
... All loops in the **test suite** consist of one or mom such statements. We define three possible results for a **compiler** attempting to vector& a **loop**. A **loop** is veczorized if the **compiler** generates vector instructions for all vectorixable statements in the **loop**. ...
Cited by 56 - Related articles - Library Search - All 9 versions

SUIF: An infrastructure for research on parallelizing and optimizing **compilers** [PDF] from psu.edu
RP Wilson, RS French, CS Wilson... - ACM Sigplan ..., 1994 - portal.acm.org
... C and SUIF, and Michael Wolf for building the initial system as well as the **loop** transformation library. ... We also want to thank John Ruttenberg for letting us use the Muttiflow **test suite**. The SUIF **compiler** project has been supported in part by DARPA contracts N00014-87-K-0828 ...
Cited by 476 - Related articles - All 27 versions

[PDF] Parallel loops - a **test suite** for parallelizing **compilers**: description and example results [PDF] from psu.edu
J Dongarra, M Furtney, S Reinhardt... - Parallel Computing, 1991 - Citeseer
... 5. **Loop** Scoring Vendors were mailed a magnetic tape containing the Parallel Loops collection. ... Thus, the use of **compiler** directives or interactive compilation features to gain additional parallelizations was ... The objective of this **test suite** has been to provide a measure of system ...
Cited by 14 - Related articles - View as HTML - All 56 versions

Idiom recognition in the Polaris parallelizing **compiler** [PDF] from psu.edu
B Pottenger - ... of the 9th international conference on ..., 1995 - portal.acm.org
... Available **compilers** typically are able to substitute the induction variable in the inner **loop** only. ... iteration of a **loop** [1 1]. There is one important case in our application **test suite** where the recognition of wrap-around **loop** bounds is a necessary precursor to the solution of an ...
Cited by 66 - Related articles - All 24 versions

Timing variation in dual **loop** benchmark [PDF] from psu.edu
N Altman, N Weiderman - ACM SIGAda Ada Letters, 1988 - portal.acm.org
... In fact, thi s dual **loop** paradigm can be found in three commonly used benchmark suites, namely the Prototype Ad a **Compiler** Evaluation **test suite** [1], the Performance Issues Working Group (PIWG) **test suite** [5] develope d by a working group of the Association for Computing ...
Cited by 21 - Related articles - Library Search - All 14 versions

The jastadd extensible java **compiler** [PDF] from ox.ac.uk
T Ekman... - Proceedings of the 22nd annual ACM SIGPLAN ..., 2007 - portal.acm.org
... with the language specification, actually passing a slightly higher number of tests in the Jacks **test suite** [jac07a] than ... Our Java **compiler** follows this implementation scheme [EH06 ... 4.3.1 The enhanced for **loop** Consider extending Java 1.4 with the enhanced for **loop** of Java 5: for ...
Cited by 129 - Related articles - BL Direct - All 10 versions

A **Test Suite** Approach for Fortran90D **Compilers** on MIMD Distributed Memory Parallel Computers
MY Wu... - Scalable High Performance Computing ..., 2002 - ieeexplore.ieee.org
... An in- troductory example of Gaussian elimination is used, among other programs in our **test suite**, to explain the compilation techniques. ... Arrays a and row are partitioned by **compiler** direc- tives. ... An array operation in the FortrangOD program is sequen- tialized into a do **loop**. ...
Cited by 10 - Related articles - All 2 versions

[BOOK] The SUIF **compiler** system: a parallelizing and optimizing research **compiler** [PDF] from stanford.edu
RP Wilson, R French, C Wilson, S Amarasinghe... - 1994 - db.stanford.edu
... C and SUIF, and Michael Wolf for building the initial system as well as the **loop** transformation library. ... We also want to thank John Ruttenberg for letting us use the Multiflow **test suite**. The SUIF **compiler** project has been supported in part by DARPA contracts N00014-87-K-0828 ...
Cited by 36 - Related articles - View as HTML - Library Search - All 10 versions

A comparison study of automatically vectorizing Fortran **compilers**
H Nobayashi... - ..., '89. Proceedings of the 1989 ACM/ ..., 2009 - ieeexplore.ieee.org
... 1. Livermore Fortran Kernels (LFK) A well-known set of 24 Fortran **loop** kernels devel oped ... 2. Argonne National Laboratory's **Test Suite** (ATS) [1] A set of 100 loops in four categories ... Dependence Analysis: the ability of a **compiler** to perform global flow analysis and dependence ...
Cited by 11 - Related articles - All 9 versions

Evaluating OpenMP performance analysis tools with the APART **test suite** [PDF] from psu.edu
M Gerndt, B Mohr... - Euro-Par 2004 Parallel Processing, 2004 - Springer
... though outer **loop** has much more iterations insufficient work in parallel **loop**: **loop** overhead dominates ... analysis tools have different thresholds/sensitivities, it is important that the **test suite** is parametrized ... A **compiler** switch pmfunc directs the **compiler** to instrument user functions ...
Cited by 15 - Related articles - BL Direct - All 11 versions

□ Create email alert

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

loop "training set" OR "test suite" coı Search

Go to Google Home - About Google - About Google Scholar

Web Images Videos Maps News Shopping Gmail more ▾ Scholar Preferences | Sign in

Google scholar modify benchmark based on feedback compile Search Advanced Scholar Search

Scholar Articles and patents ▾ anytime ▾ include citations ▾ Create email alert Results **1** - **10** of about **12,300**. (**0.03** sec)

Compiler-based prefetching for recursive data structures [PDF] from psu.edu
CK Luk, TC Mowry - ACM SIGOPS Operating Systems Review, 1996 - portal.acm.org
... If the RDS does **change** radically, the program will still behave correctly, but prefetching will not ... we performed detailed cycle-by-cycle simulations of the entire Olden **benchmark** suite [17 ... The Olden **bench- mark** suite contains ten pointer-**based** applications written in C, which are ...
Cited by 375 - Related articles - BL Direct - All 23 versions

Evaluating iterative compilation [PDF] from psu.edu
GG Fursin, MFP O'Boyle... - Languages and **Compilers** ..., 2005 - Springer
... best program version is shown for three of the six different platforms across the three **benchmarks**. ... Otherwise the current best version is retained and we see no **change** in execution time reduction ... tion time) from the SPEC **benchmark** suite in order to find a good optimisation and ...
Cited by 63 - Related articles - BL Direct - All 16 versions

Probabilistic source-level optimisation of embedded programs [PDF] from psu.edu
B Franke, M O'Boyle, J Thomson... - ..., **compilers**, and tools for ..., 2005 - portal.acm.org
... probability, but unlike the space exploring random search algorithm, probabilities can **change** over time ... 5.2 **Benchmarks** We have chosen the **UTDSP** [15, 19] **benchmark** suite to evaluate ... This set of **benchmarks** contains compute-intensive DSP kernels as well as applications ...
Cited by 41 - Related articles - BL Direct - All 19 versions

Adaptive java optimisation using instance-**based** learning [PDF] from psu.edu
S Long... - Proceedings of the 18th annual international ..., 2004 - portal.acm.org
... proach which evolves and adapts to applications and archi- tectural **change**, without sacrificing performance. ... This means that for each **benchmark**, the system has previously seen and op- timised the other fifteen **benchmarks** which act as training examples. ...
Cited by 26 - Related articles - All 10 versions

[PDF] ISPW-6 software process example [PDF] from ucl.ac.uk
M Kellner, P Feiler, A Finkelstein, T Katayama... - 1991 - eprints.ucl.ac.uk
... The use of a standard **benchmark** problem facilitates comparisons of various modeling approaches. ... **Modify** Unit Test Package 2.9.1. Description This step involves the modification of the ... Subsequent iterations of this step may be **based** upon **feedback** from testing, indicating that ...
Cited by 32 - Related articles - All 4 versions

A portable sampling-**based** profiler for Java virtual machines [PDF] from psu.edu
J Whaley - Proceedings of the ACM 2000 conference on Java ..., 2000 - portal.acm.org
... In this equation, ATo,e~an refers to the **change** in total run time, Tcompi~ refers to the amount of time it ... of sam- pie profiling: It presents a detailed performance eval- uation of the overhead and accuracy of our sampling- **based** profiler on a variety of **benchmarks** and systems. ...
Cited by 77 - Related articles - All 10 versions

[PDF] **Feedback** assisted iterative compilation [PDF] from psu.edu
MO'Boyle, PMW Knijnenburg... - Preprint, 2000 - Citeseer
... However, we can **change** this order dynamically. ... For each **benchmark** and platform, we have used two agres- sive **compiler** optimization levels. ... that Strategies 1 and 2 per- form about equally well: only small differences in speedup are found and across the **benchmarks** in some ...
Cited by 25 - Related articles - View as HTML - All 2 versions

A framework for reducing instruction scheduling overhead in dynamic **compilers**
V Tang, J Siu, A Vasilevskiy... - Proceedings of the 2006 ..., 2006 - portal.acm.org
... Register copies are required whenever a value kept in a register needs to be preserved for future use, but the current instruction will **change** the value in the register. ... On the z990 processor, we used a **benchmark** very similar to SPECjvm98. ... Page 6. 6 **Bench- mark** Rel. ...
Cited by 5 - Related articles

A heuristic search algorithm **based** on unified transformation framework [PDF] from psu.edu
S Long... - Parallel Processing, 2005. ICPP 2005. ..., 2005 - ieeexplore.ieee.org
... Sixteen code segments were chosen from two widely- used **benchmark** suites, namely Java Grande Forum ... For each **bench- mark**, the algorithm evaluated the first 100 points it reached in the ... to represent this modification, and a set of primitives are used to **modify** the polyhedron ...
Cited by 20 - Related articles - All 14 versions

[PDF] Design and experience: Using the Intel Itanium2 processor performance monitoring unit to implement **feedback** optimizations [PDF] from usc.es
Y Choi, A Knies, G Vedaraman... - EPIC2 Workshop, 2002 - dec.usc.es
... although we have not fully investigated complete combinations or individual thresholds for each **benchmark**. ... heuristics **change** the way the **compiler** schedules hot loads and their consumers, but ... 3.3 Results Figure 9 shows results from SPEC CPU2000 integer **benchmarks**. ...
Cited by 11 - Related articles - View as HTML - All 2 versions

Create email alert

Goooooooooogle ▸
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

modify benchmark based on feedba Search

Go to Google Home - About Google - About Google Scholar

Web Images Videos Maps News Shopping Gmail more ▾ Scholar Preferences | Sign in

Google scholar modify training set based on feedback compile Search Advanced Scholar Search

Scholar Articles and patents ▾ anytime ▾ include citations ▾ Create email alert Results **1** - **10** of about **15,500**. (**0.22** sec)

Adaptive java optimisation using instance-**based** learning [PDF] from psu.edu
S Long - Proceedings of the 18th annual international ..., 2004 - portal.acm.org
... proach which evolves and adapts to applications and archi- tectural **change**, without sacrificing performance. ... An alternative approach is to try many transformations on a **set** of suitably chosen programs or **training** examples. ...
Cited by 25 - Related articles - All 10 versions

MILEPOST GCC: machine learning **based** research **compiler** [PDF] from inria.fr
G Fursin, C Miranda, O Temam, M Namolaru... - 2008 - hal.inria.fr
... To verify that we can **change** the default optimization pass orders using ICI, we recompiled the same bench- mark with the -O3 flag ... Our approach to selecting good passes for programs is **based** upon the construction of a probabilistic model on a **set** of M **training** programs and ...
Cited by 25 - Related articles - All 23 versions

Instruction **based** memory distance analysis and its application [PDF] from psu.edu
C Fang, S Carr, S Onder... - 2005 - computer.org
... and translate those changes into the cache effects for a large input without using that large input **set**. ... because of the **change** in alignment of structures in a cache line with the **change** in data ... conditions is not satisfied: (1) the instruction does not occur in at least one **training** run, (2 ...
Cited by 24 - Related articles - Library Search - All 13 versions

Profile-**based** dynamic voltage and frequency scaling for a multiple clock domain microprocessor [PDF] from psu.edu
G Magklis, ML Scott, G Semeraro, DH Albonesi... - 2003 - computer.org
... The profiling-**based** cases were trained using the smaller input **set**. ... The L+F and F mech- anisms, however, will always **change** frequencies when they encounter a node that was long-running in the **training** runs, even when they reach it over a different path. ...
Cited by 117 - Related articles - BL Direct - All 31 versions

Midatasets: Creating the conditions for a more realistic evaluation of iterative optimization [PDF] from psu.edu
G Fursin, J Cavazos, MO'boyle... - ... Architectures and **Compilers**, 2007 - Springer
... and (3) evaluate iterative optimization under more "realistic" conditions where data sets **change** across executions. ... Using a data **set** different from the one used for **training** causes some degradation ... We use this data **set** suite to understand how iterative optimization behaves in a ...
Cited by 33 - Related articles - BL Direct - All 21 versions

Adaptive optimizing **compilers** for the 21st century [PDF] from psu.edu
KD Cooper, D Subramanian... - The Journal of Supercomputing, 2001 - Springer
... particularly important codes, the user may want a version that limits its **training set** to that ... However, their model included a limited **set** of transformations that attacked a single problem—cache ... Changing these parameters of the genetic algorithm do **change** its behavior, but do not ...
Cited by 148 - Related articles - BL Direct - All 15 versions

[PDF] Reuse-distance-**based** miss-rate prediction on a per instruction basis [PDF] from psu.edu
C Fang, S Carr, S Onder... - Proceedings of the first ACM SIGPLAN ..., 2004 - Citeseer
... to predict the miss rate of the same program run on the reference input data **set**. ... In 189.lucas, approximately 31% of the memory operations do not appear in both **training** runs ... These extra instructions **change** the reuse distance because differ- ent memory locations are accessed ...
Cited by 34 - Related articles - View as HTML - All 2 versions

Evaluating iterative compilation [PDF] from psu.edu
GG Fursin, MFP O'Boyle... - Languages and **Compilers** ..., 2005 - Springer
... What is re- quired is an approach which evolves and adapts to architectural **change** without sacrificing ... The Compaq **compiler** with the optimisation level **set** to -O5 becomes a high level restructurer which ... This is followed by an evaluation of the use of smaller **training** data as a ...
Cited by 59 - Related articles - BL Direct - All 16 versions

[PDF] Neural network-**based** diesel engine emissions prediction using in-cylinder combustion pressure [PDF] from atkinsonllc.com
ML Traver, RJ Atkinson... - SAE transactions, 1999 - atkinsonllc.com
... a **change** in exhaust emis- sions and when the analyzers respond to that **change**, the network is ... HC and CO have proven far more elusive in finding a **set** of input parameters that ... may partially be due to switching acquisi- tion systems between the gathering of the **training** and the ...
Cited by 41 - Related articles - View as HTML - BL Direct - All 8 versions

Compiler-Directed Cache Line Size Adaptivity [PDF] from psu.edu
D Nicolaescu, X Ji, A Veidenbaum... - Intelligent Memory ..., 2001 - Springer
... We used profiling to determine the best cache line size for each loop, we run the benchmarks using the **training** input **set**, determined for each ... the minimum miss rate and used that data to run the benchmarks using a **compiler** generated instruction to **change** the cache line ...
Cited by 8 - Related articles - BL Direct - All 13 versions

Create email alert

Goooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

modify training set based on feedba Search

Go to Google Home - About Google - About Google Scholar

ACM DIGITAL LIBRARY

Us Patent & Trademark Office

SIGN IN SIGN UP

genetic algorithm optimize compiler

Searching for: genetic algorithm optimize compiler benchmark (start a new search)
Found **296** of **1,639,151** within *The ACM Guide to Computing Literature*

Limit your search to Publications from ACM and Affiliated Organizations

REFINE YOUR SEARCH

▾ Refine by Keywords
genetic algorithm optir
Discovered Terms

▾ Refine by People
Names
Institutions
Authors
Editors
Reviewers

▾ Refine by Publications
Publication Year
Publication Names
ACM Publications
All Publications
Content Formats
Publishers

▾ Refine by Conferences
Sponsors
Events
Proceeding Series

ADVANCED SEARCH
Advanced Search

FEEDBACK
Please provide us with feedback

Found **296** of **1,639,151**

Search Results Related Journals Related Magazines Related SIGs Related Conferences

Results 1 - 20 of 296 Sort by relevance in expandec

Result page: **1** 2 3 4 5 6 7 8 9 10 n

1 Genetic programming applied to compiler heuristic optimization
Mark Stephenson, Una-May O'Reilly, Martin C. Martin, Saman Amarasinghe
April 2003 **EuroGP'03**: Proceedings of the 6th European conference on Genetic programming
Publisher: Springer-Verlag
Bibliometrics: Downloads (6 Weeks): n/a, Downloads (12 Months): n/a, Downloads (Overall): n/a, Citation

Genetic programming (GP) has a natural niche in the optimization of small but high payoff software heur
We use GP to optimize the priority functions associated with two well known compiler heuristics: predica
hyperblock formation, and register ...

2 Proceedings of the 10th annual conference on Genetic and evolutionary computation
Conor Ryan, Maarten Keijzer
July 2008 **GECCO '08**: Proceedings of the 10th annual conference on Genetic and evolutionary com
Publisher: ACM
Bibliometrics: Downloads (6 Weeks): n/a, Downloads (12 Months): n/a, Downloads (Overall): n/a, Citation

These proceedings contain the papers presented at the *10th Annual Genetic and Evolutionary Computati*
Conference (GECCO-2008), held in Atlanta, Georgia, July 12-16, 2008. GECCO has returned to the U.S.
maintains an impressive record of both ...

3 Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computatior
Conor Ryan, Maarten Keijzer
July 2008 **GECCO '08**: Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation
Publisher: ACM
Bibliometrics: Downloads (6 Weeks): n/a, Downloads (12 Months): n/a, Downloads (Overall): n/a, Citation

These proceedings contain the papers presented at the *10th Annual Genetic and Evolutionary Computati*
Conference (GECCO-2008), held in Atlanta, Georgia, July 12-16, 2008. GECCO has returned to the U.S.
maintains an impressive record of both ...

4 Finding representative workloads for computer system design
Jan Lodewijk Bonebakker
March 2007 Finding representative workloads for computer system design
Publisher: Sun Microsystems, Inc.
Full text available: Pdf (3.72 MB)
Bibliometrics: Downloads (6 Weeks): 1, Downloads (12 Months): 1, Downloads (Overall): 1, Citation Count

This work explores how improved workload characterization can be used for a better selection of represe
workloads within the computer system and processor design process. We find that metrics easily availab
modern computer systems provide ...

5 VISTA: VPO interactive system for tuning applications
Prasad Kulkarni, Wankang Zhao, Stephen Hines, David Whalley, Xin Yuan, Robert van Engelen, Kyle Galliva
Hiser, Jack Davidson, Baosheng Cai, Mark Bailey, Hwashin Moon, Kyunghwan Cho, Yunheung Paek
November 2006 **Transactions on Embedded Computing Systems (TECS)**, Volume 5 Issue 4
Publisher: ACM Request Permissions

ACM DIGITAL LIBRARY

Us Patent & Trademark Office

SIGN IN SIGN UP

compiler (benchmark OR test OR tra

Searching for: compiler (benchmark OR test OR training) (set OR suite) (start a new search)
Found **367** of **1,639,151** within *The ACM Guide to Computing Literature*

Limit your search to Publications from ACM and Affiliated Organizations

REFINE YOUR SEARCH

▾ Refine by Keywords
compiler (benchmark (
Discovered Terms

▾ Refine by People
Names
Institutions
Authors
Editors
Reviewers

▾ Refine by Publications
Publication Year
Publication Names
ACM Publications
All Publications
Content Formats
Publishers

▾ Refine by Conferences
Sponsors
Events
Proceeding Series

ADVANCED SEARCH
Advanced Search

FEEDBACK
Please provide us with feedback

Found **367** of **1,639,151**

Search Results Related Journals Related Magazines Related SIGs Related Conferences

Results 1 - 20 of 367 Sort by relevance in expa

Result page: **1** 2 3 4 5 6 7 8 9 1(

1 Meta optimization: improving compiler heuristics with machine learning
Mark Stephenson, Saman Amarasinghe, Martin Martin, Una-May O'Reilly
June 2003 **PLDI '03:** Proceedings of the ACM SIGPLAN 2003 conference on Programming language desig implementation
Publisher: ACM Request Permissions
Full text available: Pdf (302.23 KB)
Bibliometrics: Downloads (6 Weeks): 7, Downloads (12 Months): 72, Downloads (Overall): 888, Citation Cc

Compiler writers have crafted many heuristics over the years to approximately solve NP-hard problems e Finding a heuristic that performs well on a broad range of applications is a tedious and difficult process. introduces Meta Optimization, ...

Keywords: compiler heuristics, genetic programming, machine learning, priority functions

Also published in:
May 2003 **SIGPLAN Notices** Volume 38 Issue 5

2 Genetic programming applied to compiler heuristic optimization
Mark Stephenson, Una-May O'Reilly, Martin C. Martin, Saman Amarasinghe
April 2003 **EuroGP'03:** Proceedings of the 6th European conference on Genetic programming
Publisher: Springer-Verlag
Bibliometrics: Downloads (6 Weeks): n/a, Downloads (12 Months): n/a, Downloads (Overall): n/a, Citation

Genetic programming (GP) has a natural niche in the optimization of small but high payoff software heur use GP to optimize the priority functions associated with two well known compiler heuristics: predicated formation, and register ...

3 Evidence-based static branch prediction using machine learning
Brad Calder, Dirk Grunwald, Michael Jones, Donald Lindsay, James Martin, Michael Mozer, Benjamin Zorn
January 1997 **Transactions on Programming Languages and Systems (TOPLAS)**, Volume 19 I
Publisher: ACM Request Permissions
Full text available: Pdf (515.50 KB)
Bibliometrics: Downloads (6 Weeks): 9, Downloads (12 Months): 60, Downloads (Overall): 564, Citation Cc

Correctly predicting the direction that branches will take is increasingly important in today's wide-issue c architectures. The name program-based branch prediction is given to static branch prediction techniques their ...

Keywords: branch prediction, decision trees, machine learning, neural networks, performance evaluatic optimization

4 Collective optimization: A practical collaborative approach
Grigori Fursin, Olivier Temam
December 2010 **Transactions on Architecture and Code Optimization (TACO)**, Volume 7 Issu
Publisher: ACM Request Permissions
Full text available: PDF (1.66 MB)
Bibliometrics: Downloads (6 Weeks): 52, Downloads (12 Months): 52, Downloads (Overall): 52, Citation Cc

ACM DIGITAL LIBRARY

Us Patent & Trademark Office

SIGN IN SIGN UP

compiler loop (benchmark OR test O

Searching for: compiler loop (benchmark OR test OR training) (set OR suite) (start a new search)
Found **269** of **1,639,151** within *The ACM Guide to Computing Literature*

Limit your search to Publications from ACM and Affiliated Organizations

REFINE YOUR SEARCH

▾ Refine by Keywords

compiler loop (benchm

Discovered Terms

▾ Refine by People

Names
Institutions
Authors
Editors
Reviewers

▾ Refine by Publications

Publication Year
Publication Names
ACM Publications
All Publications
Content Formats
Publishers

▾ Refine by Conferences

Sponsors
Events
Proceeding Series

ADVANCED SEARCH

Advanced Search

FEEDBACK

Please provide us with feedback

Found **269** of **1,639,151**

Search Results Related Journals Related Magazines Related SIGs Related Conferences

Results 1 - 20 of 269

Sort by relevance in expa

Result page: **1** 2 3 4 5 6 7 8 9 10

1 Evidence-based static branch prediction using machine learning
Brad Calder, Dirk Grunwald, Michael Jones, Donald Lindsay, James Martin, Michael Mozer, Benjamin Zorn
January 1997 **Transactions on Programming Languages and Systems (TOPLAS)** , Volume 19 I
Publisher: ACM Request Permissions
Full text available Pdf (515 50 KB)
Bibliometrics Downloads (6 Weeks) 9, Downloads (12 Months): 60, Downloads (Overall) 564, Citation Cc

Correctly predicting the direction that branches will take is increasingly important in today's wide-issue c
architectures. The name program-based branch prediction is given to static branch prediction techniques
their ...

Keywords: branch prediction, decision trees, machine learning, neural networks, performance evaluatic
optimization

2 Collective optimization: A practical collaborative approach
Grigori Fursin, Olivier Temam
December 2010 **Transactions on Architecture and Code Optimization (TACO)** , Volume 7 Issu
Publisher: ACM Request Permissions
Full text available PDF (1.66 MB)
Bibliometrics Downloads (6 Weeks): 52, Downloads (12 Months): 52, Downloads (Overall) 52, Citation Cc

Iterative optimization is a popular and efficient research approach to optimize programs using feedback-
compilation. However, one of the key limitations that prevented widespread use in production compilers
day practice is the necessity ...
Keywords: Collective optimization, adaptive compiler, collective optimization database, continuous opti
function cloning, iterative compilation, multiple datasets, program characterization, program reaction to
runtime adaptation, self-tuning computing systems, statistical optimization

3 Automatic performance model construction for the fast software exploration of new hardware desic
John Cavazos, Christophe Dubach, Felix Agakov, Edwin Bonilla, Michael F. P. O'Boyle, Grigori Fursin, Olivier
October 2006 **CASES '06:** Proceedings of the 2006 international conference on Compilers, architecture and
for embedded systems
Publisher: ACM
Full text available Pdf (254 09 KB)
Bibliometrics Downloads (6 Weeks) 7, Downloads (12 Months): 35, Downloads (Overall) 249, Citation Cc

Developing an optimizing compiler for a newly proposed architecture is extremely difficult when there is
simulator of the machine available. Designing such a compiler requires running many experiments in orc
understand how different optimizations ...

Keywords: architecture, artificial neural networks, compiler optimization, machine learning, performanc

4 Value-based clock gating and operation packing: dynamic strategies for improving processor powe
performance
David Brooks, Margaret Martonosi
May 2000 **Transactions on Computer Systems (TOCS)** , Volume 18 Issue 2
Publisher: ACM Request Permissions
Full text available Pdf (210 51 KB)
Bibliometrics Downloads (6 Weeks) 5, Downloads (12 Months) 49, Downloads (Overall) 704, Citation Cc

IEEE Xplore®

IEEE

SEARCH RESULTS

You searched for: (tune OR tailor OR select) compiler (option OR heuristic OR directive)

Results per Page 25

Showing 1 - 10 of 10 results

Compiler optimization-space exploration

Triantafyllis, S., Vachharajani, M., Vachharajani, N., August, D.I.,
Code Generation and Optimization, 2003. CGO 2003. International Symposium on
Digital Object Identifier: 10.1109/CGO.2003.1191546
Publication Year: 2003 , Page(s): 204 - 215

IEEE CONFERENCES

Combining models and guided empirical search to optimize for multiple levels of the memory hierarchy

Chen, C.; Chame, J.; Hall, M.;
Code Generation and Optimization, 2005. CGO 2005. International Symposium on
Digital Object Identifier: 10.1109/CGO.2005.10
Publication Year: 2005 , Page(s): 111 - 122

IEEE CONFERENCES

Automatic selection of GCC optimization options using a gene weighted genetic algorithm

San-Chih Lin; Chi-Kuang Chang; Nai-Wei Lin;
Computer Systems Architecture Conference, 2008. ACSAC 2008. 13th Asia-Pacific
Digital Object Identifier: 10.1109/APCSAC.2008.4625477
Publication Year: 2008 , Page(s): 1 - 8

IEEE CONFERENCES

An overview of the ECO project

Chame, J.; Chun Chen; Diniz, P.; Hall, M.; Yoon-Ju Lee; Lucas, R.F.;
Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International
Digital Object Identifier: 10.1109/IPDPS.2006.1639571
Publication Year: 2006

IEEE CONFERENCES

Scheduling Tasks with Resource Requirements in Hard Real-Time Systems

Wei Zhao; Ramamritham, K.; Stankovic, J.A.;
Software Engineering, IEEE Transactions on
Volume: SE-13 , Issue: 5
Digital Object Identifier: 10.1109/TSE.1987.233201
Publication Year: 1987 , Page(s): 564 - 577

IEEE JOURNALS

Annotation-based empirical performance tuning using Orio

Hartono, A.; Norris, B.; Sadayappan, P.;
Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on
Digital Object Identifier: 10.1109/IPDPS.2009.5161004
Publication Year: 2009 , Page(s): 1 - 11

IEEE CONFERENCES

On the use of query-driven XML auto-indexing

Schmidt, K. ; Harder, T. ;
Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on
Digital Object Identifier: 10.1109/ICDEW.2010.5452741
Publication Year: 2010 , Page(s): 81 - 86

IEEE CONFERENCES

Adaptive tuning in a dynamically changing resource environment

Seyong Lee, Eigenmann, R. ;
Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on
Digital Object Identifier: 10.1109/IPDPS.2008.4536099
Publication Year: 2008 , Page(s): 1 - 5

IEEE CONFERENCES

Spatial Based Feature Generation for Machine Learning Based Optimization Compilation

Malik, Abid M. ;
Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on
Digital Object Identifier: 10.1109/ICMLA.2010.147
Publication Year: 2010 , Page(s): 925 - 930

IEEE CONFERENCES

Heuristic tradeoffs between latency and energy consumption in register assignment

Anand, R. ; Jacome, M. ; De Veciana, G. ;
Hardware/Software Codesign, 2000. CODES 2000. Proceedings of the Eighth International Workshop on
Publication Year: 2000 , Page(s): 115 - 119

IEEE CONFERENCES

IEEE Xplore® DIGITAL LIBRARY

IEEE

SEARCH RESULTS

You searched for: (benchmark OR training) (set OR suite) loop

You refined by:

Publication Year: 1967 - 2005

Results per Page 25 Showing 1 - 25 of 656 results

Unroll-and-jam using uniformly generated sets

Carr, S. ; Yiping Guan,
Microarchitecture, 1997. Proceedings., Thirtieth Annual IEEE/ACM International Symposium on
Digital Object Identifier: 10.1109/MICRO.1997.645832
Publication Year: 1997 , Page(s): 349 - 357

IEEE CONFERENCES

Long haul participation in a distributed interactive simulation demonstration

Woodyard, J.M. ; Reif, D.C. ,
Aerospace and Electronics Conference, 1995. NAECON 1995., Proceedings of the IEEE 1995 National
Volume: 2
Digital Object Identifier: 10.1109/NAECON.1995.522030
Publication Year: 1995 , Page(s): 810 - 816 vol.2

IEEE CONFERENCES

Multigrain parallel processing on OSCAR CMP

Kimura, K. ; Kodaka, T. ; Obata, M. ; Kasahara, H. ,
Innovative Architecture for Future Generation High-Performance Processors and Systems, 2003
Digital Object Identifier: 10.1109/IWIA.2003.1262783
Publication Year: 2003 , Page(s): 56 - 65

IEEE CONFERENCES

Stage scheduling: a technique to reduce the register requirements of a module schedule

Eichenberger, A.E. ; Davidson, E.S. ,
Microarchitecture, 1995. Proceedings of the 28th Annual International Symposium on
Digital Object Identifier: 10.1109/MICRO.1995.476840
Publication Year: 1995 , Page(s): 338 - 349

IEEE CONFERENCES

An implementation of interprocedural bounded regular section analysis

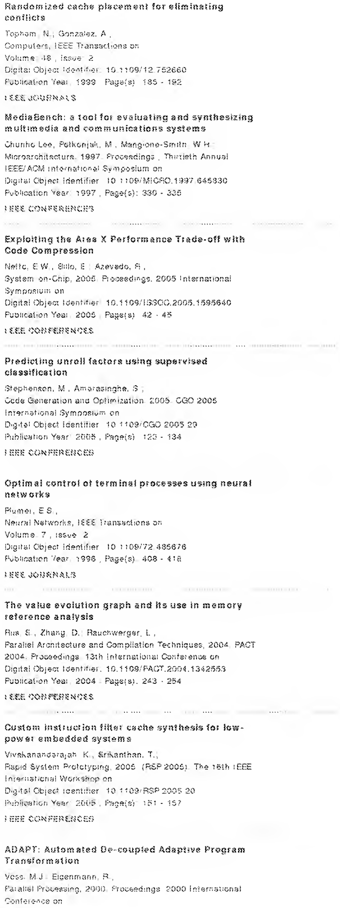
Havlak, P. ; Kennedy, K. ,
Parallel and Distributed Systems, IEEE Transactions on
Volume: 2 , Issue: 3
Digital Object Identifier: 10.1109/71.86110
Publication Year: 1991 , Page(s): 350 - 360

IEEE JOURNALS

Static methods in hybrid branch prediction

Grunwald, D. ; Lindsay, D. ; Zorn, B. ;
Parallel Architectures and Compilation Techniques, 1998. Proceedings. 1998 International Conference on
Digital Object Identifier: 10.1109/PACT.1998.727254
Publication Year: 1998 , Page(s): 222 - 229

IEEE CONFERENCES



Randomized cache placement for eliminating conflicts

Topham, N. , Gonzalez, A. ,
Computers, IEEE Transactions on
Volume: 48 , Issue: 2
Digital Object Identifier: 10.1109/12.752660
Publication Year: 1999 , Page(s): 185 - 192

IEEE JOURNALS

MediaBench: a tool for evaluating and synthesizing multimedia and communications systems

Chunho Lee, Potkonjak, M. , Mangione-Smith, W.H. ,
Microarchitecture, 1997. Proceedings., Thirtieth Annual IEEE/ACM International Symposium on
Digital Object Identifier: 10.1109/MICRO.1997.645830
Publication Year: 1997 , Page(s): 330 - 335

IEEE CONFERENCES

Exploiting the Area X Performance Trade-off with Code Compression

Netto, E.W. , Billo, E. , Azevedo, R. ,
System-on-Chip, 2005. Proceedings. 2005 International Symposium on
Digital Object Identifier: 10.1109/ISSOC.2005.1595640
Publication Year: 2005 , Page(s): 42 - 45

IEEE CONFERENCES

Predicting unroll factors using supervised classification

Stephenson, M. , Amarasinghe, S. ,
Code Generation and Optimization, 2005. CGO 2005. International Symposium on
Digital Object Identifier: 10.1109/CGO.2005.29
Publication Year: 2005 , Page(s): 123 - 134

IEEE CONFERENCES

Optimal control of terminal processes using neural networks

Plumer, E.S. ,
Neural Networks, IEEE Transactions on
Volume: 7 , Issue: 2
Digital Object Identifier: 10.1109/72.485676
Publication Year: 1996 , Page(s): 408 - 418

IEEE JOURNALS

The value evolution graph and its use in memory reference analysis

Rus, S. , Zhang, D. , Rauchwerger, L. ,
Parallel Architecture and Compilation Techniques, 2004. PACT 2004. Proceedings. 13th International Conference on
Digital Object Identifier: 10.1109/PACT.2004.1342553
Publication Year: 2004 , Page(s): 243 - 254

IEEE CONFERENCES

Custom instruction filter cache synthesis for low-power embedded systems

Vivekanandarajah, K. , Srikanthan, T. ,
Rapid System Prototyping, 2005. (RSP 2005). The 16th IEEE International Workshop on
Digital Object Identifier: 10.1109/RSP.2005.20
Publication Year: 2005 , Page(s): 151 - 157

IEEE CONFERENCES

ADAPT: Automated De-coupled Adaptive Program Transformation

Voss, M.J. , Eigenmann, R. ,
Parallel Processing, 2000. Proceedings. 2000 International Conference on

Digital Object Identifier: 10.1109/ICPP.2000.876107
Publication Year: 2000 , Page(s): 163 - 170

IEEE CONFERENCES

A benchmark study approach to fault diagnosis of industrial process control systems

Patton, R.;
Control Loop Assessment and Diagnosis, 2005. The IEE Seminar on (Ref. No. 2005/11008)
Digital Object Identifier: 10.1049/ic:20050175
Publication Year: 2005 , Page(s): 61 - 79

IET CONFERENCES

Compiler support for parallel code generation through kernel recognition

Arenaz, M.; Tourino, J.; Doallo, R.;
Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International
Digital Object Identifier: 10.1109/IPDPS.2004.1303015
Publication Year: 2004

IEEE CONFERENCES

Swing module scheduling: a lifetime-sensitive approach

Llosa, J.; Gonzalez, A.; Ayguade, E.; Valero, M.;
Parallel Architectures and Compilation Techniques, 1996., Proceedings of the 1996 Conference on
Digital Object Identifier: 10.1109/PACT.1996.554030
Publication Year: 1996 , Page(s): 80 - 86

IEEE CONFERENCES

Back propagation simulations using limited precision calculations

Holt, J.L.; Baker, T.E.;
Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on
Volume: ii
Digital Object Identifier: 10.1109/IJCNN.1991.155324
Publication Year: 1991 , Page(s): 121 - 126 vol.2

IEEE CONFERENCES

Lifetime-sensitive modulo scheduling in a production environment

Llosa, J.; Ayguade, E.; Gonzalez, A.; Valero, M.; Eckhardt, J.;
Computers, IEEE Transactions on
Volume: 50 , Issue: 3
Digital Object Identifier: 10.1109/12.910814
Publication Year: 2001 , Page(s): 234 - 249

IEEE JOURNALS

Sequential network construction for time series prediction

Cholewo, T.J.; Zurada, J.M.;
Neural Networks,1997., International Conference on
Volume: 4
Digital Object Identifier: 10.1109/ICNN.1997.614214
Publication Year: 1997 , Page(s): 2034 - 2038 vol.4

IEEE CONFERENCES

Capacity control in classifiers for pattern recognition

Solla, S.A.;
Neural Networks for Signal Processing [1992] II., Proceedings of the 1992 IEEE-SP Workshop
Digital Object Identifier: 10.1109/NNSP.1992.253687
Publication Year: 1992 , Page(s): 255 - 266

IEEE CONFERENCES

A characteristic-point-based fuzzy inference system aimed to minimize the number of fuzzy rules

Tang-Kai Yin;
Fuzzy Systems, IEEE Transactions on
Volume: 12 , Issue: 2
Digital Object Identifier: 10.1109/TFUZZ.2004.825088
Publication Year: 2004 , Page(s): 260 - 273

IEEE JOURNALS

An information-theoretic measure to evaluate data partitions in multiple classifiers

Dara, R.A. ; Makrehchi, M. ; Kamel, M. ;
Systems, Man and Cybernetics, 2004 IEEE International Conference on
Volume: 5
Digital Object Identifier: 10.1109/ICSMC.2004.1401295
Publication Year: 2004 , Page(s): 4826 - 4831 vol.5

IEEE CONFERENCES

Self-Organizing Gaussian Fuzzy CMAC with Truth Value Restriction

Nguyen, M.N. ; Shi, D. ; Quek, C. ;
Information Technology and Applications, 2005. ICITA 2005. Third International Conference on
Volume: 2
Digital Object Identifier: 10.1109/ICITA.2005.250
Publication Year: 2005 , Page(s): 185 - 190

IEEE CONFERENCES

Efficient techniques for advanced data dependence analysis

Kyriakopoulos, K. ; Psarris, K. ;
Parallel Architectures and Compilation Techniques, 2005. PACT 2005. 14th International Conference on
Digital Object Identifier: 10.1109/PACT.2005.19
Publication Year: 2005 , Page(s): 143 - 153

IEEE CONFERENCES

IEEE Xplore®
DIGITAL LIBRARY

IEEE

SEARCH RESULTS

You searched for: compiler optimization (benchmark OR training)

Results per Page 25

Showing 1 - 12 of 12 results

Aestimo: a feedback-directed optimization evaluation tool

Berube, P.; Amaral, J.N.;
Performance Analysis of Systems and Software, 2006 IEEE International Symposium on
Digital Object Identifier: 10.1109/ISPASS.2006.1620809
Publication Year: 2006 , Page(s): 251 - 260

IEEE CONFERENCES

Workload Reduction for Multi-input Feedback-Directed Optimization

Berube, P., Amaral, J.N., Ho, R., Silvera, R.,
Code Generation and Optimization, 2009. CGO 2009. International Symposium on
Digital Object Identifier: 10.1109/CGO.2009.23
Publication Year: 2009 , Page(s): 59 - 69

IEEE CONFERENCES

On the impact of data input sets on statistical compiler tuning

Haneda, M., Knijnenburg, P.M.W., Wijshoff, H.A.G.,
Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International
Digital Object Identifier: 10.1109/IPDPS.2006.1639724
Publication Year: 2006

IEEE CONFERENCES

Automatic Program Segment Similarity Detection in Targeted Program Performance Improvement

Wu, H.; Park, E.; Kaplarevic, M.; Zhang, Y.; Bolat, M.; Li, X.; Gao, G.R.;
Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International
Digital Object Identifier: 10.1109/IPDPS.2007.370642
Publication Year: 2007 , Page(s): 1 - 8

IEEE CONFERENCES

Predicting unroll factors using supervised classification

Stephenson, M., Amarasinghe, S.,
Code Generation and Optimization, 2005. CGO 2005. International Symposium on
Digital Object Identifier: 10.1109/CGO.2005.29
Publication Year: 2005 , Page(s): 123 - 134

IEEE CONFERENCES

A Lightweight Iterative Compilation Approach for Optimization Parameter Selection

Yonggang Che, Zhenghua Wang,
Computer and Computational Sciences, 2006. IMSCCS '06. First International Multi-Symposiums on
Volume: 1
Digital Object Identifier: 10.1109/IMSCCS.2006.11
Publication Year: 2006 , Page(s): 318 - 325

IEEE CONFERENCES

On the predictability of program behavior using different input data sets

Wei Chung Hsu, Howard Chen, Pen Chung Yew, Dong-Yuan Chen,
Interaction between Compilers and Computer Architectures, 2002. Proceedings. Sixth Annual Workshop on
Digital Object Identifier: 10.1109/INTERA.2002.995842
Publication Year: 2002 , Page(s): 45 - 53

IEEE CONFERENCES

Using Support Vector Machines to Learn How to Compile a Method

Sanchez, R.N.; Amaral, J.N.; Szafron, D.; Pirvu, M.; Stoodley, M.;
Computer Architecture and High Performance Computing (SBAC-PAD), 2010 22nd International Symposium on
Digital Object Identifier: 10.1109/SBAC-PAD.2010.35
Publication Year: 2010 , Page(s): 223 - 230

IEEE CONFERENCES

The accuracy of initial prediction in two-phase dynamic binary translators

Wu, Y.; Breternitz, M.; Quek, J.; Etzion, O.; Fang, J.;
Code Generation and Optimization, 2004. CGO 2004. International Symposium on
Digital Object Identifier: 10.1109/CGO.2004.1281677
Publication Year: 2004 , Page(s): 227 - 238

IEEE CONFERENCES

A solution to the can or cannot problem of learning based compilation

Shun Long; Wei-Heng Zhu;
Natural Computation (ICNC), 2010 Sixth International Conference on
Volume: 6
Digital Object Identifier: 10.1109/ICNC.2010.5583919
Publication Year: 2010 , Page(s): 3261 - 3265

IEEE CONFERENCES

Outlier Detection for Learning-Based Optimizing Compiler

Shun Long; Weiheng Zhu;
Frontier of Computer Science and Technology (FCST), 2010 Fifth International Conference on
Digital Object Identifier: 10.1109/FCST.2010.31
Publication Year: 2010 , Page(s): 570 - 575

IEEE CONFERENCES

Reality-based optimization

McFarling, S.;
Code Generation and Optimization, 2003. CGO 2003. International Symposium on
Digital Object Identifier: 10.1109/CGO.2003.1191533
Publication Year: 2003 , Page(s): 59 - 68

IEEE CONFERENCES